IEIIT-CNR

Randomized Algorithms for Systems and Control: Theory and Applications

Roberto Tempo

IEIIT-CNR
Politecnico di Torino
roberto.tempo@polito.it

IEIIT-CNR

- Additional documents, papers, etc, please consult
  http://staff.polito.it/roberto.tempo/
- Questions may be sent to
  roberto.tempo@polito.it

IEIIT-CNR

References

- R. Tempo, G. Calafiore and F. Dabbene, "Randomized Algorithms for Analysis and Control of Uncertain Systems," Springer-Verlag, London, 2005
- R. Tempo and H. Ishii, "Monte Carlo and Las Vegas Randomized Algorithms for Systems and Control: An Introduction," European Journal of Control, Vol. 13, pp. 189-203, 2007
- RACT: Randomized Algorithms Control Toolbox for Matlab

http://ract.sourceforge.net

IEIIT-CNR

Overview

- Preliminaries
- Randomized Algorithms for Analysis
- Probabilistic Robust Synthesis
- Randomized Algorithms for Optimal Control (LQR)
- Extensions
- Applications: Probabilistic Control of Mini UAVs

IEIIT-CNR

Preliminaries

IEIIT-CNR

Randomized Algorithms (RAs)

- Randomized algorithms are frequently used in many areas of engineering, computer science, physics, finance, optimization,…but their appearance in systems and control is mostly limited to Monte Carlo simulations…
- Main objective of this mini-course: Introduction to rigorous study of RAs for uncertain systems and control, with specific applications

Report Documentation Page			*Form Approved OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.			
1. REPORT DATE **MAY 2008**	2. REPORT TYPE		3. DATES COVERED **00-00-2008 to 00-00-2008**
4. TITLE AND SUBTITLE **Randomized Algorithms for Systems and Control: Theory and Applications**			5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)			5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **IEIIT-CNR,Politecnico di Torino,Corso Duca degli Abruzzi ,24 - 10129 Torino Italy,**			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**			
13. SUPPLEMENTARY NOTES **See also ADM002223. Presented at the NATO/RTO Systems Concepts and Integration Panel Lecture Series SCI-195 on Advanced Autonomous Formation Control and Trajectory Management Techniques for Multiple Micro UAV Applications held in Glasgow, United Kingdom on 19-21 May 2008.**			
14. ABSTRACT			
15. SUBJECT TERMS			

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**31**	

IEIIT-CNR

Randomized Algorithms (RAs)

- Combinatorial optimization, computational geometry
- Examples: Data structuring, search trees, graph algorithms, sorting (RQS), …
- Motion and path planning problems
- Mathematics of finance: Computation of path integrals
- Bioinformatics (string matching problems)

IEIIT-CNR

Uncertainty

- Uncertainty has been always a critical issue in control theory and applications
- First methods to deal with uncertainty were based on a stochastic approach
- Optimal control: LQG and Kalman filter
- Since early 80's alternative deterministic approach (worst-case or robust) has been proposed

IEIIT-CNR

Robustness

- Major stepping stone in 1981: Formulation of the $\mathcal{H}_\infty$ problem by George Zames
- Various "robust" methods to handle uncertainty now exist: Structured singular values, Kharitonov, optimization-based (LMI), *l*-one optimal control, quantitative feedback theory (QFT)

IEIIT-CNR

Robustness

- Late 80's and early 90's: Robust control theory became a well-assessed area
- Successful industrial applications in aerospace, chemical, electrical, mechanical engineering, …
- However, …

IEIIT-CNR

Limitations of Robust Control - 1

- Researchers realized some drawbacks of robust control
- Consider uncertainty Δ bounded in a set $\mathcal{B}$ of radius ρ. Largest value of ρ such that the system is stable for all $\Delta \in \mathcal{B}$ is called (worst-case) robustness margin
- Conservatism: Worst case robustness margin may be small
- Discontinuity: Worst case robustness margin may be discontinuous wrt problem data

IEIIT-CNR

Limitations of Robust Control - 2

- Computational Complexity: Worst case robustness is often $\mathcal{NP}$-hard (not solvable in polynomial time unless $\mathcal{P} = \mathcal{NP}$)[1]
- Various robustness problems are $\mathcal{NP}$-hard
  - static output feedback
  - structured singular value
  - stability of interval matrices

[1] V. Blondel and J.N. Tsitsiklis (2000)

Conservatism and Complexity Trade-Off

IEIIT-CNR

- Uncertain or control design parameters often enter into the system in a nonlinear/nonconvex fashion
- To avoid complexity issues (or just to find a solution of the problem) relaxation techniques such as SOS are used
- Study issues about the accuracy of the approximation introduced and related complexity

Different Paradigm Proposed

IEIIT-CNR

- New paradigm proposed is based on uncertainty randomization and leads to randomized algorithms for analysis and synthesis
- Within this setting a different notion of problem tractability is needed
- Objective: Breaking the curse of dimensionality[1]

[1] R. Bellman (1957)

Probability *and* Robustness

IEIIT-CNR

- The interplay of Probability and Robustness for control of uncertain systems
- Robustness: Deterministic uncertainty bounded
- Probability: Random uncertainty (pdf is known)
- Computation of the probability of performance
- Controller which stabilizes *most* uncertain systems

Key Features

IEIIT-CNR

- We obtain larger robustness margins at the expense of a small risk
- We study the probability degradation *beyond* the robustness margins
- Computational complexity is generally not an issue: Randomized algorithms are low complexity

Uncertain Systems

IEIIT-CNR

$M(s)$ System	Δ Uncertainty

- Δ belongs to a structured set $\mathcal{B}_D$
  - Parametric uncertainty q
  - Nonparametric uncertainty Δ_i
  - Mixed uncertainty

Worst Case Model

IEIIT-CNR

- Worst case model: Set membership uncertainty
- The uncertainty Δ is bounded in a set $\mathcal{B}_D$

$$\Delta \in \mathcal{B}_D$$

- Real parametric uncertainty $q=[q_1,\dots, q_\ell] \in \mathbb{R}^\ell$

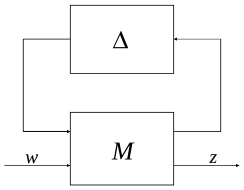
$$q_i \in [q_i^-, q_i^+]$$

- Nonparametric uncertainty

$$\Delta_i \in \{\Delta_i \in \mathbb{R}^{n,n} : \|\Delta_i\| \le 1\}$$

Robustness

IEIIT-CNR



- Uncertainty Δ is bounded in a structured set $\mathcal{B}_D$
- $z = \mathcal{F}_u(M,\Delta)\, w$, where $\mathcal{F}_u(M,\Delta)$ is the upper LFT

Objective of Robustness

IEIIT-CNR

- Objective of robustness: To guarantee stability and performance for all

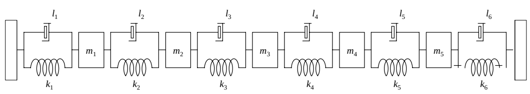
$$\Delta \in \mathcal{B}_D$$

- Different probabilistic paradigm based on uncertainty randomization of Δ within $\mathcal{B}_D$

Example: Flexible Structure - 1

IEIIT-CNR

- Mass spring damper model
- Real parametric uncertainty affecting stiffness and damping
- Complex unmodeled dynamics (nonparametric)



Flexible Structure - 2

IEIIT-CNR

- M-Δ configuration for controlled systems and study stability of

$$M(s) = C(sI - A)^{-1}B$$

$$\Delta = \begin{bmatrix} q_1 I_5 & 0 & 0 \\ 0 & q_2 I_5 & 0 \\ 0 & 0 & \Delta_1 \end{bmatrix}$$

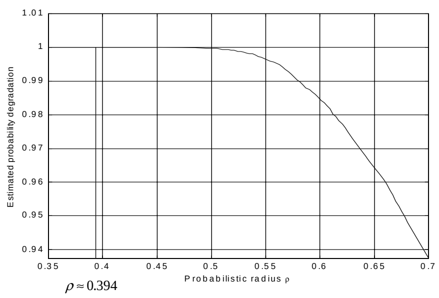
$q_1, q_2 \in \mathbb{R}$

$\Delta_1 \in \mathbb{C}^{4,4}$

$\Delta \in \mathcal{B}_D = \{\Delta \in \mathcal{D} : \bar{\sigma}(\Delta) < \rho\}$

Probability Degradation Function

IEIIT-CNR



Probabilistic Model

IEIIT-CNR

- Probability density function associated to $\mathcal{B}_D$
- We now assume that Δ is a random matrix with given density function $f_\Delta(\Delta)$ and support $\mathcal{B}_D$
- Example: Δ is uniform in $\mathcal{B}_D$

IEIIT-CNR

Uniform Density

- Take $f_\Delta(\Delta)=\mathcal{U}[\mathcal{B}_D]$ (uniform density within $\mathcal{B}_D$)

$$\mathcal{U}[\mathcal{B}_D]=\begin{cases}\dfrac{1}{vol(\mathcal{B}_D)} & if\ \Delta\in\mathcal{B}_D\\ 0 & otherwise\end{cases}$$

- In this case, for a subset $\mathcal{S}\subseteq\mathcal{B}_D$

$$\Pr\{\Delta\in\mathcal{S}\}=\frac{\int_{\mathcal{S}}d\Delta}{vol(\mathcal{B}_D)}=\frac{vol(\mathcal{S})}{vol(\mathcal{B}_D)}$$

IEIIT-CNR

Performance Function

- In classical robustness we guarantee that a certain performance requirement is attained for all $\Delta\in\mathcal{B}_D$
- This can be stated in terms of a performance function

$$J = J(\Delta)$$

- Examples: $\mathcal{H}_\infty$ performance and robust stability

IEIIT-CNR

Example: $\mathcal{H}_\infty$ Performance - 1

- Compute the $\mathcal{H}_\infty$ norm of the upper LFT $\mathcal{F}_u(M,\Delta)$

$$J(\Delta) = \| \mathcal{F}_u(M, \Delta)\|_\infty$$

- For given $\gamma>0$, check if

$$J(\Delta) < \gamma$$

for all Δ in $\mathcal{B}_D$

IEIIT-CNR

Example: $\mathcal{H}_\infty$ Performance - 2

- Continuous time SISO systems with real parametric uncertainty q with upper LFT

$$\mathcal{F}_u(M,\Delta) = \mathcal{F}_u(M,q) =$$

$$\frac{0.5q_1q_2s+10^{-5}q_1}{(10^{-5}+0.05q_2)s^2+(0.00102+0.5q_2)s+(2\cdot 10^{-5}+0.5q_1^2)}$$

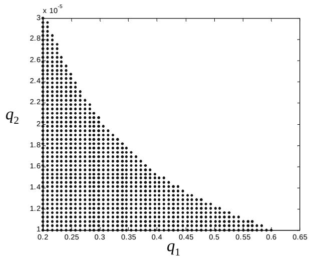
where $q_1\in[0.2, 0.6]$ and $q_2\in[10^{-5},3\cdot 10^{-5}]$

- Letting $J(q) = \| \mathcal{F}_u(M,q)\|_\infty$, we choose γ=0.003
- Check if $J(q)<\gamma$ for all q in these intervals

IEIIT-CNR

Example: $\mathcal{H}_\infty$ Performance - 3

- The set of q_1, q_2 for which $J(q)<\gamma$ is shown below



IEIIT-CNR

Example[1]: Robust Stability - 1

- Consider the closed loop uncertain polynomial

$$p(s,q) =$$

$$(1+r^2+6q_1+6q_2+2q_1q_2)+(q_1+q_2+3)s+(q_1+q_2+1)s^2+s^3$$

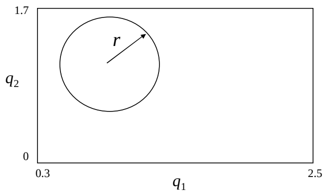
where $q_1\in[0.3, 2.5]$, $q_2\in[0,1.7]$ and r=0.5

- Check stability for all q in these intervals

[1] G. Truxal (1961)

Example: Robust Stability - 2

- Set of unstable polynomials



- Taking r=0 the unstable set reduces to a singleton

Good and Bad Sets

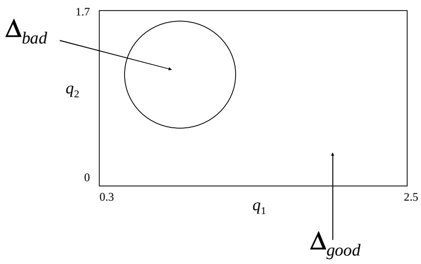
- We define two subsets of $\mathcal{B}_D$

$$\Delta_{good} = \{\Delta: J(\Delta) \le \gamma\} \subseteq \mathcal{B}_D$$
$$\Delta_{bad} = \{\Delta: J(\Delta) > \gamma\} \subseteq \mathcal{B}_D$$

- Δ_{good} is the set of Δ's satisfying performance
- Measure of robustness is

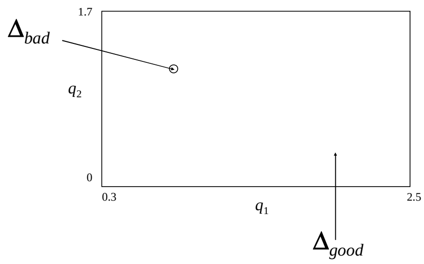
$$vol(\Delta_{good}) = \int_{\Delta_{good}} d\Delta$$

Example of Good and Bad Sets



Example of Good and Bad Sets - 2

Taking small r



Probabilistic Robustness Measure

- In worst-case analysis we compute γ such that all Δ satisfy performance. Equivalently, we evaluate γ such that

$$\Delta_{good} = \mathcal{B}_D$$

- In a probabilistic setting, we are satisfied if the ratio

$$\frac{vol(\Delta_{good})}{vol(\mathcal{B}_D)}$$

is close to one

Probability of Performance[1]

- We define the probability of performance as

$$p_\gamma = \Pr\{J(\Delta) \le \gamma\}$$

- Notice that, if $f_\Delta(\Delta)$ is uniform, then

$$p_\gamma = \frac{vol(\Delta_{good})}{vol(\mathcal{B}_D)}$$

[1] R.F. Stengel (1980)

IEIIT-CNR

Example: Closed-Form Computation

- For Truxal's example, we compute p_γ in closed-form
- For uniform distribution, we have

$$vol(\Delta_{good}) = 3.74 - \pi\, r^2$$

$$vol(\mathcal{B}_D) = 3.74$$

IEIIT-CNR

P1: Performance Verification

- For given performance level γ, check whether

$$J(\Delta) \leq \gamma$$

for all Δ in $\mathcal{B}_D$

- Compute the probability of performance p_γ

IEIIT-CNR

P2: Worst-Case Performance

- Find $J_{\max}$ such that

$$J_{\max} = \max_{\Delta \in \mathcal{B}_D} J(\Delta)$$

- Compute the worst case performance (or its probabilistic counterpart)

IEIIT-CNR

Randomized Algorithms for Analysis

IEIIT-CNR

Randomized Algorithm: Definition

- Randomized Algorithm (RA): An algorithm that makes random choices during its execution to produce a result
- Example of a "random choice" is a coin toss

heads or *tails*

IEIIT-CNR

Randomized Algorithm: Definition

- Randomized Algorithm (RA): An algorithm that makes random choices during its execution to produce a result
- For hybrid systems, "random choices" could be switching between different states or logical operations
- For uncertain systems, "random choices" require (vector or matrix) random sample generation

Monte Carlo Randomized Algorithm

- Monte Carlo Randomized Algorithm (MCRA): A randomized algorithm that may produce incorrect results, but with bounded error probability

Las Vegas Randomized Algorithm

- Las Vegas Randomized Algorithm (LVRA): A randomized algorithm that always produces correct results, the only variation from one run to another is the running time

Randomization of Uncertain Systems

- Consider random uncertainty Δ, associated pdf and bounding set B
- Δ is a (real or complex) random vector (parametric uncertainty) or matrix (nonparametric uncertainty)
- Consider a performance function

$$J(\Delta): B \to \mathbf{R}$$

  and level $\gamma > 0$
- Define worst case and average performance

$$J_{\max} = \max_{\Delta \in B} J(\Delta) \qquad J_{\text{ave}} = E_{\Delta}(J(\Delta))$$

Example: H_∞ Performance - 1

- H_∞ performance of sensitivity function

$$B = \{\Delta: \Delta = \text{bdiag}\,(\Delta_1,\dots,\Delta_q) \in \mathbf{F}^{n,m},\ \sigma_{max}(\Delta) \le \rho\}$$

$$S(s,\Delta) = 1/(1 + P(s,\Delta)\,C(s))$$

$$J(\Delta) = \|S(s,\Delta)\|_\infty$$

Example: H_∞ Performance - 2

- H_∞ performance of sensitivity function

$$B = \{\Delta: \Delta = \text{bdiag}\,(\Delta_1,\dots,\Delta_q) \in \mathbf{F}^{n,m},\ \sigma_{\max}(\Delta) \le \rho\}$$

$$S(s,\Delta) = 1/(1 + P(s,\Delta)\,C(s))$$

$$J(\Delta) = \|S(s,\Delta)\|_\infty$$

- Objective: Check if

$$J_{\max} \leqslant \gamma \quad \text{and} \quad J_{\text{ave}} \leqslant \gamma$$

- These are uncertain decision problems

Two Problem Instances

- We have two problem instances for worst case performance

$$J_{\max} \leqslant \gamma \ \text{ and } \ J_{\max} > \gamma$$

  and two problem instances for average case performance

$$J_{\text{ave}} \leqslant \gamma \ \text{ and } \ J_{\text{ave}} > \gamma$$

- This leads to one-sided and two-sided MCRA

IEIIT-CNR

One-Sided MCRA

- One-sided MCRA: Always provide a correct solution in one of the instances (they may provide a wrong solution in the other instance)
- Consider the empirical maximum

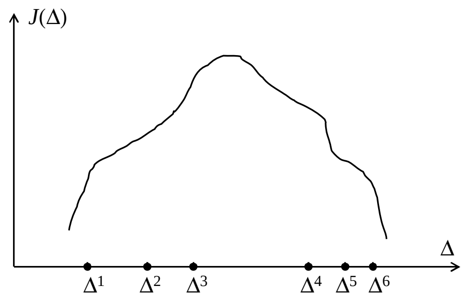
$$\hat{J}_{\max} = \max_{i=1,\ldots,N} J(\Delta^i)$$

where Δ^i are random samples and N is the sample size

- Check if $\hat{J}_{\max} \leqslant \gamma$ or $\hat{J}_{\max} > \gamma$

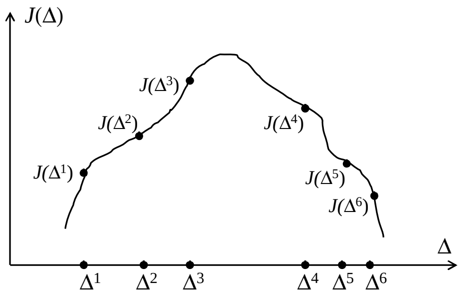
IEIIT-CNR

One-Sided MCRA: Case 1



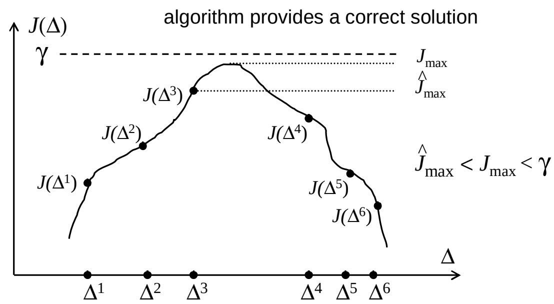
IEIIT-CNR

One-Sided MCRA: Case 1



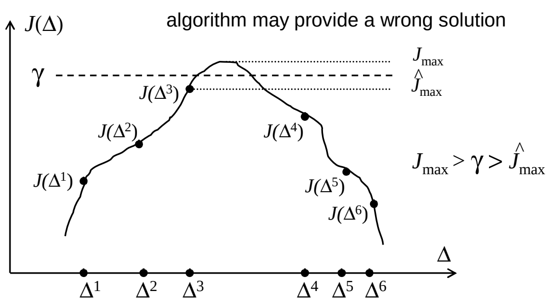
IEIIT-CNR

One-Sided MCRA: Case 1



IEIIT-CNR

One-Sided MCRA: Case 2



IEIIT-CNR

Two-Sided MCRA

- Two-sided MCRA: They may provide a wrong solution in both instances
- Consider the empirical average

$$\hat{J}_{\mathrm{ave}} = \operatorname*{ave}_{i=1,\ldots,N} J(\Delta^i)$$

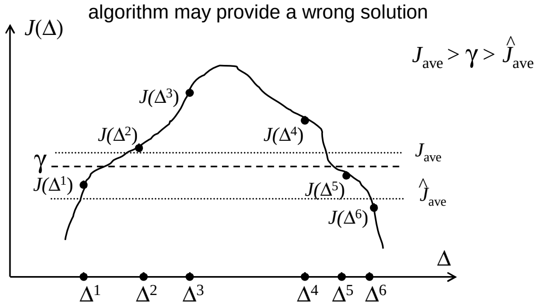
where Δ^i are random samples and N is the sample size

- Check if $\hat{J}_{\mathrm{ave}} \leqslant \gamma$ or $\hat{J}_{\mathrm{ave}} > \gamma$

IEIIT-CNR

Two-Sided MCRA: Case 1

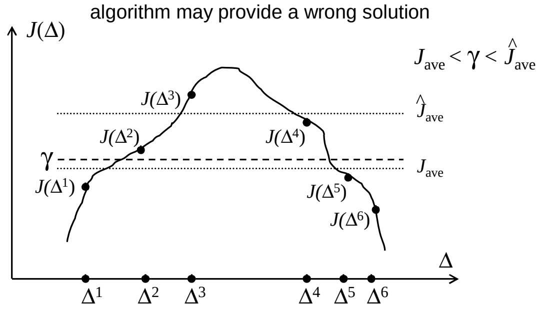
algorithm may provide a wrong solution



IEIIT-CNR

Two-Sided MCRA: Case 2

algorithm may provide a wrong solution



IEIIT-CNR

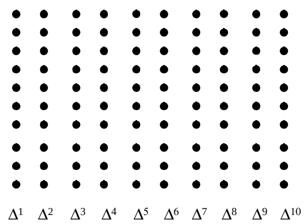
Las Vegas Randomized Algorithms

- We also have zero-sided (Las Vegas) randomized algorithms
- Las Vegas Randomized Algorithm (LVRA): Always give the correct solution
- The solution obtained with a LVRA is probabilistic, so "always" means with probability one
- Running time may be different from one run to another
- We can study the *average* running time

IEIIT-CNR

Discrete Bounding Set

Consider random uncertainty Δ, a *discrete* bounding set B, given pdf and performance function



IEIIT-CNR

The Las Vegas Viewpoint

- Consider discrete random variables
- The sample space is discrete and M^N possible choices can be made
- In the binary case we have 2^N
- Finding maximum requires ordering the 2^N choices
- Las Vegas can be used for ordering real numbers

- Example: Randomized Quick Sort for sorting real numbers (classical in computer science)

IEIIT-CNR

Ingredients for RAs

- Assume that Δ is random with pdf $f_\Delta(\Delta)$ with support $\mathcal{B}_D$
- Accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$ be assigned
- Performance function for analysis and level

↓ ↓

$J = J(\Delta)$ γ

IEIIT-CNR

Randomized Algorithms for Analysis

- Two classes of randomized algorithms for probabilistic robust performance analysis

- P1: Performance verification (compute p_γ)
- P2: Worst-case performance (compute $J_{\max}$)

- Both are based on uncertainty randomization of Δ
- Bounds on the sample size are obtained

IEIIT-CNR

Randomized Algorithms - 2

- We estimate p_γ by means of a randomized algorithm
- First, we generate N i.i.d. samples

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_D$$

according to the density f_Δ

- We evaluate $J(\Delta^1), J(\Delta^2), \ldots, J(\Delta^N)$

IEIIT-CNR

Empirical Probability

- Construct an indicator function

$$I(\Delta^i) = \begin{cases} 1 & \mathit{if}\ J(\Delta^i) \le \gamma \\ 0 & \mathit{otherwise} \end{cases}$$

- An estimate of p_γ is the empirical probability

$$\hat{p}_N = \frac{1}{N}\sum_{i=1}^{N} I(\Delta^i) = \frac{N_{good}}{N}$$

where N_{good} is the number of samples such that $J(\Delta^i) \le \gamma$

IEIIT-CNR

A Reliable Estimate

- The empirical probability is a reliable estimate if

$$|p_\gamma - \hat{p}_N| = |\Pr\{J(\Delta) \le \gamma\} - \hat{p}_N| \le \varepsilon$$

- Find the minimum N such that

$$\Pr\{|p_\gamma - \hat{p}_N| \le \varepsilon\} \ge 1 - \delta$$

where $\varepsilon \in (0,1)$ and $\delta \in (0,1)$

IEIIT-CNR

Chernoff Bound[1]

- For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

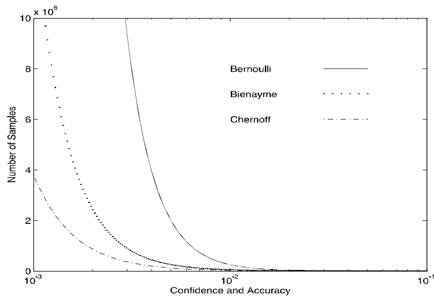
$$N \ge \frac{\log \frac{2}{\delta}}{2\varepsilon^2}$$

then $$\Pr\{|p_\gamma - \hat{p}_N| \le \varepsilon\} \ge 1 - \delta$$

[1] H. Chernoff (1952)

IEIIT-CNR

Comparison Between Bounds



Chernoff Bound

IEIIT-CNR

- Remark: Chernoff bound improves upon other bounds such as Bernoulli (Law of Large Numbers)
- Dependence on $1/\delta$ is logarithmic
- Dependence on $1/\varepsilon$ is quadratic

ε	0.1%	0.1%	0.5%	0.5%
$1\text{-}\delta$	99.9%	99.5%	99.9%	99.5%
N	$3.9\cdot10^6$	$3.0\cdot10^6$	$1.6\cdot10^6$	$1.2\cdot10^5$

Computational Complexity of RAs

IEIIT-CNR

- RAs are efficient (polynomial-time) because

1. Random sample generation of Δ^i can be performed in polynomial-time
2. Cost associated with the evaluation of $J(\Delta^i)$ for fixed Δ^i is polynomial-time
3. Sample size is polynomial in the problem size and probabilistic levels ε and δ

1. Random Sample Generation

IEIIT-CNR

- Random number generation (RNG): Linear and nonlinear methods for uniform generation in [0,1) such as Fibonacci, feedback shift register, BBS, MT, …
- Non-uniform univariate random variables: Suitable functional transformations (e.g., the inversion method)
- The problem is much harder: Multivariate generation of samples of Δ with pdf $f_\Delta(\Delta)$ and support $\mathcal{B}_D$
- It can be resolved in polynomial-time

2. Cost of Checking Stability

IEIIT-CNR

- Consider a polynomial

$$p(s,a) = a_0 + a_1 s + \cdots + a_n s^n$$

- To check left half plane stability we can use the Routh test. The number of multiplications needed is

$$\frac{n^2}{4} \text{ for } n \text{ even} \qquad \frac{n^2-1}{4} \text{ for } n \text{ odd}$$

- The number of divisions and additions is equal to this number
- We conclude that checking stability is $O(n^2)$

3. Bounds on the Sample Size

IEIIT-CNR

- Chernoff bound is independent on the size of $\mathcal{B}_D$, on the structure $\mathcal{D}$ on the number of blocks, on the pdf $f_\Delta(\Delta)$
- It depends only on δ and ε
- Same comments can be made for other bounds such as Bernoulli

Worst-Case Performance

IEIIT-CNR

- Recall that

$$J_{\max} = \max_{\Delta\in\mathcal{B}_D} J(\Delta)$$

- Generate N i.i.d. samples

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_D$$

according to the density f_Δ

- Compute the empirical maximum

$$\hat{J}_{\max} = \max_{i=1,\ldots,N} J(\Delta^i)$$

Worst-Case Bound (Log-over-Log)[1]

- For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\varepsilon}}$$

then

$$\Pr\left\{\Pr\left\{J(\Delta) > \hat{J}_N\right\} \leq \varepsilon\right\} \geq 1-\delta$$

[1] R. Tempo, E. W. Bai and F. Dabbene (1996)

Comparison and Comments

- Number of samples is much smaller than Chernoff
- Bound is a specific instance of the fpras (fully polynomial randomized approximated scheme) theory
- Dependence on $1/\varepsilon$ is basically linear $\left(\log \frac{1}{1-\varepsilon} \approx \varepsilon\right)$

ε	0.1%	0.1%	0.5%	0.5%	0.01%	0.001%
1-δ	99.9%	99.5%	99.9%	99.5%	99.99%	99.999%
N	$6.91 \cdot 10^3$	$5.30 \cdot 10^3$	$1.38 \cdot 10^3$	$1.06 \cdot 10^3$	$9.21 \cdot 10^4$	$1.16 \cdot 10^6$

Volumetric Interpretation

- In the case of $f_\Delta(\Delta)$ uniform, we have

$$\Pr\left\{J(\Delta) > \hat{J}_N\right\} = \frac{vol(\Delta_{bad})}{vol(\mathcal{B}_D)}$$

- Therefore

$$\Pr\left\{\Pr\left\{J(\Delta) > \hat{J}_N\right\} \leq \varepsilon\right\} \geq 1-\delta$$

is equivalent to

$$\Pr\left\{vol(\Delta_{bad}) \leq \varepsilon\, vol(\mathcal{B}_D)\right\} \geq 1-\delta$$

Confidence Intervals

- The Chernoff and worst-case bounds can be computed *a-priori* and provide an explicit functional relation

$$N = N(\varepsilon, \delta)$$

- The sample size obtained with the confidence intervals is not explicit
- Given $\delta \in (0,1)$, upper and lower confidence intervals p_L and p_U are such that

$$\Pr\{p_L \leq p_\gamma \leq p_U\} = 1-\delta$$

Confidence Intervals - 2

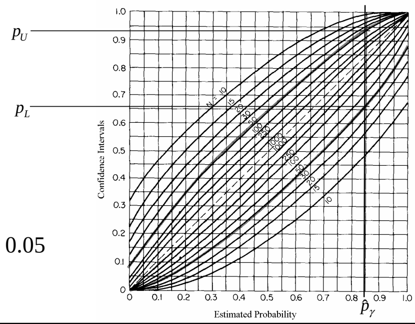
- The probabilities p_L and p_U can be computed *a posteriori* when the value of N_{good} is known, solving equations of the type

$$\sum_{k=N_{good}}^{N} \binom{N}{k} p_L^k (1-p_L)^{N-k} = \delta_L$$

$$\sum_{k=0}^{N_{good}} \binom{N}{k} p_U^k (1-p_U)^{N-k} = \delta_U$$

with $\delta_L + \delta_U = \delta$

Confidence Intervals - 3



$\delta = 0.05$

IEIIT-CNR

Statistical Learning Theory

- The Chernoff Bound studies the problem

$$\Pr\left\{\left|p_\gamma - \hat{p}_N\right| \le \varepsilon\right\} \ge 1-\delta$$

where $p_\gamma = \Pr\{J(\Delta) \le \gamma\}$

- Performance function J is fixed
- Statistical Learning Theory computes bounds on the sample size for the problem

$$\Pr\left\{\left|\Pr(J(\Delta)\le\gamma) - \hat{p}_N\right| \le \varepsilon, \forall J \in \mathcal{J}\right\} \ge 1-\delta$$

where $\mathcal{J}$ is a given class of functions

IEIIT-CNR

VC and P-dimension[1,2]

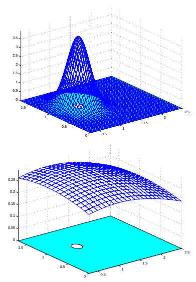
- Statistical Learning Theory aims at studying uniform Law of Large Numbers
- The bounds obtained depend on quantities called VC-dimension (if J is a binary valued function), or P-dimension (if J is a continuous valued function)
- VC and P-dimension are measures of the problem complexity

[1] M. Vidyasagar (1997)
[2] E.D. Sontag (1998)

IEIIT-CNR

Choice of the Distribution - 1

- The probability $\Pr\{\Delta \in \mathcal{S}\}$ depends on $f_\Delta(\Delta)$
- It may vary between 0 and 1 depending on the pdf $f_\Delta(\Delta)$



IEIIT-CNR

Choice of the Distribution - 2

- The bounds discussed are independent on the choice of the distribution but for computing $\Pr\{J(\Delta) \le \gamma\}$ we need to know the distribution $f_\Delta(\Delta)$
- Some research has been done in order to find the worst-case distribution in a certain class[1]
- Uniform distribution is the worst-case if a certain target is convex and centrally symmetric

[1] B. R. Barmish and C. M. Lagoa (1997)

IEIIT-CNR

Choice of the Distribution - 3

- Minimax properties of the uniform distribution have been studied[1]

[1] E. W. Bai, R. Tempo and M. Fu (1998)

IEIIT-CNR

Probabilistic Robust Synthesis

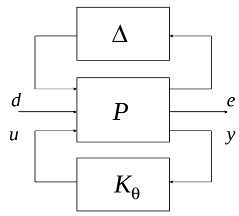
IEIIT-CNR

Analysis vs Design with Uncertainty

- Starting point: Worst-case analysis versus design
- Consider an interval family $p(s,q)$, $q \in \mathcal{B}_q = \{q \in \mathbb{R}^n, \|q\|_\infty \le 1\}$
- Analysis problem:
  – Check if $p(s,q)$ is stable for all $q \in \mathcal{B}_q$
  Answer: Kharitonov Theorem
- Design Problem:
  – Does there exist a $q \in \mathcal{B}_q$ such that $p(s,q)$ is stable?
  Answer: *Unknown* in general

IEIIT-CNR

Synthesis Paradigm



- Design the parameterized controller K_θ to guarantee stability and performance

IEIIT-CNR

Synthesis Performance Function

- Recall that the parameterized controller is K_θ
- We replace $J(\Delta)$ with a synthesis performance function

$$J = J(\Delta,\theta)$$

where $\theta \in \Theta$ represents the controller parameters to be determined and their bounding set

IEIIT-CNR

Randomized Algorithms for Synthesis

- Two classes of RAs for probabilistic synthesis
- Average performance synthesis[1]
- Based on expected value minimization
- Use of Statistical Learning Theory results
- Very general problems can be handled
- Existing bounds are very conservative and controller randomization is required
- Ongoing research aiming at major reduction of sample size

[1] M. Vidyasagar (1998)

IEIIT-CNR

Randomized Algorithms for Synthesis

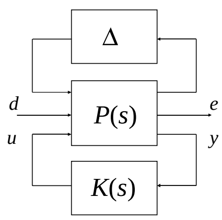
- Robust performance synthesis[1]
- Problem reformulation as robust feasibility
- Only convex problems can be handled
- Finite-time convergence with probability one is obtained

[1]B. Polyak and R. Tempo (2001)

IEIIT-CNR

Robust Performance Synthesis

- Uncertainty randomization of Δ in $\mathcal{B}_\mathcal{D}$
- Convex optimization to design the controller $K(s)$



IEIIT-CNR

RAs for Optimal Control (LQR)

IEIIT-CNR

Uncertain Systems in State Space

- We consider a state space description of the uncertain system

$$\dot{x}(t) = A(\Delta)x(t) + Bu(t)$$

with $x(0)=x_0$; $x\in\mathbb{R}^n$; $u\in\mathbb{R}^m$, $\Delta\in \mathcal{B}_D$

- For example, $A(\Delta)$ is an interval matrix with bounded entries $a_{ij}^- \le a_{ij} \le a_{ij}^+$

IEIIT-CNR

Interval and Vertex Matrices

- We consider interval uncertainty $\mathcal{A}$ (i.e. when $\Delta\in \mathcal{B}_D$)
- That is, a_{ik} ranges in the interval for all i, k

$$|a_{ik} - a_{ik}^*| \le w_{ik}$$

where a_{ik}^* are nominal values and w_{ik} are weights

- Define the $N = 2^{n^2}$ vertex matrices $A^1, A^2,\ldots, A^N$

$$a_{ik} = a_{ik}^* + w_{ik} \quad \text{or} \quad a_{ik} = a_{ik}^* - w_{ik}$$

for all $i, k = 1, 2, \ldots, n$

IEIIT-CNR

Common Lyapunov Functions

- Given matrices A^*, W and feedback K, find a *common quadratic Lyapunov* function $Q > 0$ for the system

$$\dot{x}(t) = (A + B\,K)\,x(t) \quad \text{for all} \quad A\in \mathcal{A}$$

- Find $Q > 0$ such that

$$L(Q, A) = (A+BK)^T Q + Q\,(A+BK) < 0 \quad \text{for all} \quad A \in \mathcal{A}$$

- Equivalently, find $Q > 0$ such that

$$\lambda_{max} L(Q, A) < 0 \quad \text{for all} \quad A \in \mathcal{A}$$

IEIIT-CNR

Lyapunov Stability of Interval Systems

- Quadratic Lyapunov stability analysis and synthesis of interval systems are NP-hard problems
- In principle, they can be solved in one-shot with convex optimization, but the number of constraints is exponential
- We can use relaxation (e.g. $\pi/2$ Theorem[1]) or randomization

[1] Yu. Nesterov (1997)

IEIIT-CNR

Vertex Solution

- Due to convexity, it suffices to study $L(Q, A) < 0$ for all vertex matrices[1]
- Question: Do we really need to check all the vertex matrices ($N = 2^{n^2}$)?

[1] H.P. Horisberger, P.R. Belanger (1976)

IEIIT-CNR

Vertex Reduction

- Answer: It suffices to check "only" a subset of 2^{2n} vertex matrices[1]
- This is still exponential (the problem is NP-hard), but it leads to a major computational improvement for medium size problems (e.g. n = 8 or 10)
- For example, for n=8, N is of the order 10^5 (instead of 10^{19})

[1] T. Alamo, R. Tempo, D. Rodriguez, E.F. Camacho (2007)

IEIIT-CNR

Diagonal Matrices and Generalizations

- Transform the original problem from full square matrices A to diagonal matrices $Z \in \mathbf{R}^{2n,2n}$
- It suffices to check the vertices of Z
- Extensions for L_2-gain minimization and other related LMI problems
- Generalizations for multiaffine interval systems

IEIIT-CNR

Las Vegas Randomized Algorithm

- We may perform randomization of the $N = 2^{n^2}$ vertices (in the worst case)
- If we select the vertices in random order according to a given pdf, we have a LVRA

IEIIT-CNR

Probabilistic Solution

- Randomly generate $A^1,\ldots, A^N$. Then, check if the Lyapunov equation

$$A^iQ + Q(A^i)^T \leq 0$$

is feasible for i=1,…,N and find a common solution $Q=Q^T>0$

- Critical problem: Even if N is relatively small, this is a hard computational problem

IEIIT-CNR

Sequential Algorithm

- Key point: Sequential algorithm which deals with one constraint at each step
- At step k we have
  Phase 1: Uncertainty randomization of Δ
  Phase 2: Gradient algorithm and projection
- Final result: Find a solution $Q=Q^T>0$ with probability one in a finite number of steps

IEIIT-CNR

Definition

- Let $\mathcal{E}_n$ be an Euclidean space

$$\mathcal{E}_n = \left\{ A = A^T \in \mathbb{R}^n, \|A\| = \sqrt{\sum_{i,k=1}^{n} a_{1k}^2} \right\}$$

and C be the cone of positive semi-definite matrices

$$C = \{A \in \mathcal{E}_n : A \geq 0\}$$

IEIIT-CNR

Projection on a Cone

- For any real symmetric matrix A we define the projection $[A]^+ \in C$ as

$$[A]^+ = \arg\min_{X\in C} \|A - X\|$$

- The projection can be computed through the eigenvalue decomposition $A=T\Lambda T^T$
- Then

$$[A]^+ = T\Lambda^+ T^T$$

where $\lambda_i^+ = \max\{\lambda_i, 0\}$

IEIIT-CNR

Phase 1: Uncertainty Randomization

- Uncertainty randomization: Generate $\Delta^k \in \mathcal{B}_D$
- Then, for guaranteed cost we obtain the Lyapunov equation

$$A(\Delta^k)Q + QA^T(\Delta^k) \le 0$$

IEIIT-CNR

Matrix Valued Function

- Define a matrix valued function

$$V(Q,\Delta^k) = A(\Delta^k)Q + QA^T(\Delta^k)$$

and a scalar function

$$v(Q,\Delta^k) = \left\| [V(Q,\Delta^k)]^+ \right\|$$

where $\|\cdot\|$ is the Frobenius norm

- We can also take the maximum eigenvalue of $V(Q,\Delta^k)$

IEIIT-CNR

Phase 2: Gradient Algorithm

- We write

$$Q^{k+1} = \begin{cases} \left[Q^k - \mu^k \partial_Q \{v(Q^k,\Delta^k)\}\right]^+ & \text{if } v(Q^k,\Delta^k) > 0 \\ Q^k & \text{otherwise} \end{cases}$$

where ∂_Q is the subgradient and the stepsize μ^k is

$$\mu^k = \frac{v(Q^k,\Delta^k) + r\left\|\partial_Q\{v(Q^k,\Delta^k)\}\right\|}{\left\|\partial_Q\{v(Q^k,\Delta^k)\}\right\|^2}$$

and r>0 is a parameter

IEIIT-CNR

Closed-form Gradient Computation

- The function $v(Q,\Delta^k)$ is convex in Q and its subgradient can be easily computed in a closed form

IEIIT-CNR

Theorem[1]

- Assumption: Every open subset of $\mathcal{B}_D$ has positive measure
- Theorem: A solution Q, if it exists, is found in a finite number of steps with probability one
- Idea of proof: The distance of Q^k from the solution set decreases at each correction step

[1] B.T. Polyak and R. Tempo (2001)

IEIIT-CNR

Example[1]

- We study a multivariable example for the design of a controller for the lateral motion of an aircraft.
- The model consists of four states and two inputs

$$\dot{x}(t)=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_{\dot\beta}(g/V) & N_p & N_\beta+N_{\dot\beta}Y_\beta & N_r-N_{\dot\beta} \end{bmatrix} x(t)+\begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ 0.035 & 0 \\ -2.53 & 0.31 \end{bmatrix} u(t)$$

[1] B.D.O. Anderson and J.B. Moore (1971)

IEIIT-CNR

Example - 2

- The state variables are
  - x_1 bank angle
  - x_2 derivative of bank angle
  - x_3 sideslip angle
  - x_4 jaw rate
- The control inputs are
  - u_1 rudder deflection
  - u_2 aileron deflection

IEIIT-CNR

Example - 3

- Nominal values: L_p=-2.93, L_β=-4.75, L_r=0.78, g/V=0.086, Y_β=-0.11, $N_{\dot\beta}$=0.1, N_p=-0.042, N_β=2.601, N_r=-0.29
- Perturbed matrix $A(\Delta)$: each parameter can take values in a range of ±15% of the nominal value
- Quadratic stability (γ=0): take $R=I$ and $S=0.01I$
- Remark: $A(\Delta)$ is multiaffine in the uncertain parameters: quadratic stability can be ascertained solving simultaneously 2^9=512 LMIs

IEIIT-CNR

Example - 4

- Sequential algorithm:
  - Initial point Q_0 randomly selected
  - 800 random matrices Δ^k
  - The algorithm converged to

$$Q=\begin{bmatrix} 0.7560 & -0.0843 & 0.1645 & 0.7338 \\ -0.0843 & 1.0927 & 0.7020 & 0.4452 \\ 0.1645 & 0.7020 & 0.7798 & 0.7382 \\ 0.7338 & 0.4452 & 0.7382 & 1.2162 \end{bmatrix}$$

IEIIT-CNR

Example - 5

- The corresponding controller

$$K=B^TQ^{-1}=\begin{bmatrix} 38.6191 & -4.3731 & 43.1284 & -49.9587 \\ -2.8814 & -10.1758 & 10.2370 & -0.4954 \end{bmatrix}$$

  satisfies all the 512 vertex LMIs and therefore it is also a quadratic stabilizing controller in a deterministic sense
- The optimal LQ controller computed on the nominal plant satisfies only 240 vertex LMIs

IEIIT-CNR

Extensions

IEIIT-CNR

Related Literature and Extensions

- Minimization of a measure of violation for problems that are not strictly feasible[1]
- Uncertainty in the control matrix, $B=B(\Delta)$, $\Delta \in \mathcal{B}_D$

  We take the feedback law

$$u = YQ^{-1}x$$

  where Y and $Q=Q^T>0$ are design variables

[1] B.R. Barmish and P. Shcherbakov (1999)

IEIIT-CNR

Related Literature

- Related literature on optimization and adaptive control with linear constraints[1,2,3,4]
- Stochastic approximation algorithms have been widely studied in the stochastic control and optimization literature[6,7]

[1] S. Agmon (1954)
[2] T.S. Motzkin and I.J. Schoenberg (1954)
[3] B.T. Polyak (1964)
[4] V.A. Bondarko and V.A. Yakubovich (1992)

[6] H.J. Kushner and G.G. Yin (2003)
[7] J.C. Spall (2003)

IEIIT-CNR

Subsequent Research

- Design of common Lyapunov functions for switched systems[1]
- From common to piecewise Lyapunov functions[2]
- Ellipsoidal algorithm instead of gradient algorithm[3]
- Stopping rule which provides the number of steps[4]
- Other algorithms have been recently proposed[5-6]

[1] D. Liberzon and R. Tempo (2004)
[2] H. Ishii, T. Basar and R. Tempo (2005)
[3] S. Kanev, B. De Schutter and M. Verhaegen (2002)
[4] Y. Oishi and H. Kimura (2003)
[5] Y. Fujisaki and Y. Oishi (2007)
[6] T. Alamo, R. Tempo, D. R. Ramirez and E. F. Camacho (2007)

IEIIT-CNR

Optimization Problems[1]

- Extensions to optimization problems
- Consider convex function $f(x)$ and function $g(x,\Delta)$ convex in x for fixed Δ
- Semi-infinite (nonlinear) programming problem

$$\min f(x)$$
$$g(x,\Delta) \le 0 \text{ for all } \Delta \in \mathcal{B}$$

- Reformulation as stochastic optimization
- Drawback: Convergence results are only asymptotic

[1] V. B. Tadic, S. P. Meyn and R. Tempo (2003)

IEIIT-CNR

Scenario Approach

- The scenario approach for convex problems[1]
- Non-sequential method which provides a one-shot solution for general convex problems
- Randomization of $\Delta \in \mathcal{B}$ and solution of a single convex optimization problem
- Derivation of a bound on the sample size[1]
- A new improved bound based on a pack-based strategy[2]

[1] G. Calafiore and M. Campi (2004)
[2] T. Alamo, R. Tempo and E.F. Camacho (2007)

IEIIT-CNR

Convex Semi-Infinite Optimization

- The semi-infinite optimization problem is

$$\min c^T\theta \quad \text{subject to } f(\theta, \Delta) \le 0 \quad \text{for all } \Delta \in \mathcal{B}$$

  where $f(\theta, \Delta) \le 0$ is convex in θ for all $\Delta \in \mathcal{B}$
- We assume that this problem is either unfeasible or, if feasible, it attains a unique solution for all $\Delta \in \mathcal{B}$ (this assumption is technical and may be removed)
- We assume that $\theta \in \Theta \subseteq \mathbf{R}^n$

IEIIT-CNR

Scenario Problem

- Using randomization, we construct a scenario problem
- Taking random samples Δ^i, $i = 1, 2, \ldots, N$, we construct

$$f(\theta, \Delta^i) \le 0, \quad i = 1, 2, \ldots, N$$

and

$$\min c^T \theta \quad \text{subject to } f(\theta, \Delta^i) \le 0, \quad i = 1, 2, \ldots, N$$

IEIIT-CNR

Theorem[1]

- Theorem: For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

$$N \ge \left\lceil 2/\varepsilon \log(1/\delta) + 2n + 2n/\varepsilon \log(2/\varepsilon) \right\rceil$$

then, with probability no smaller than 1- δ

- either the scenario problem is unfeasible and then also the semi-infinite optimization problem is unfeasible
- or, the scenario problem is feasible, then its optimal solution $\hat{\theta}_N$ satisfies

$$\Pr\{ \Delta \in \mathcal{B} : f(\theta, \Delta) > 0 \} \le \varepsilon$$

[1] G. Calafiore and M. Campi (2004)

IEIIT-CNR

A New Improved Bound[1]

- A new improved bound (based on a so-called pack-based strategy) has been recently obtained

$$N \ge \left\lceil 2/\varepsilon \log(1/2\delta) + 2n + 2n/\varepsilon \log 4 \right\rceil$$

- The main difference with the previous bound is that the factor

$$2n/\varepsilon \log(2/\varepsilon)$$

is replaced with

$$2n/\varepsilon \log 4$$

[1] T. Alamo, R. Tempo and E.F. Camacho (2007)

IEIIT-CNR

RACT

IEIIT-CNR

RACT

- RACT: Randomized Algorithms Control Toolbox for Matlab
- RACT has been developed at IEIIT-CNR and at the Institute for Control Sciences-RAS, based on a bilateral international project
- Members of the project

Andrey Tremba (Main Developer and Maintainer)
Giuseppe Calafiore
Fabrizio Dabbene
Elena Gryazina
Boris Polyak (Co-Principal Investigator)
Pavel Shcherbakov
Roberto Tempo (Co-Principal Investigator)

IEIIT-CNR

RACT

- Main features
- Define a variety of uncertain objects: scalar, vector and matrix uncertainties, with different pdfs
- Easy and fast sampling of uncertain objects of almost any type
- Randomized algorithms for probabilistic performance verification and probabilistic worst-case performance
- Randomized algorithms for feasibility of uncertain LMIs using stochastic gradient, ellipsoid or cutting plane methods (YALMIP needed)

Applications of Randomized Algorithms

Application of RAs

- Randomized algorithms have been developed for various specific applications
- Control of flexible structures
- Stability and robustness of high speed networks
- Stability of quantized sampled-data systems
- Brushless DC motors
- Control design of Mini UAV

Probabilistic Control of Mini-UAVs[1]

[1] L. Lorefice, B. Pralio and R. Tempo (2007)

Italian National Project for Fire Prevention

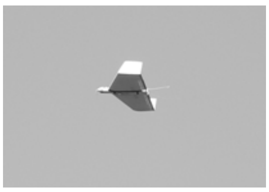
- This activity is supported by the Italian Ministry for Research within the National Project

Study and development of a real-time land control and monitoring system for fire prevention

- Five research groups are involved together with a government agency for fire surveillance and patrol located in Sicily
- The aerial platform is based on the MicroHawk configuration, developed at the Aerospace Engineering Department, Politecnico di Torino, Italy

MH1000 Platform - 1

- Platform features
  - wingspan 3.28 ft (1 m)
  - total weight 3.3 lb (1.5 kg)



MH1000 Platform - 2

- Main on-board equipment
  - various sensors and two cameras (color and infrared)
- DC motor
- Remote piloting and autonomous flight
- Flight endurance of about 40 min
- Flight envelope
  - min/max velocity: 33 ft/s (10 m/s) – 66 ft/s (17 m/s)
  - average velocity: 43 ft/s (14 m/s)

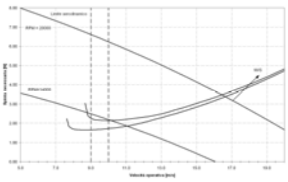
Flight Envelope (Limits)

IEIIT-CNR

Wing loading effect → total weight — Aerodynamic constraint (red) → minimum flight speed (stall effect)

Propeller sizing effect — Propulsive constraint (blu) → maximum flight speed

velocity: 33 ft/s (10 m/s) – 66 ft/s (17 m/s)



Basic on-board Systems

IEIIT-CNR

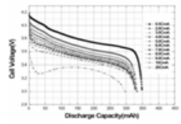
DC motor: Hacker B20-15L (4:1)
- weight: 58 g
- dimensions: Ø 20 x 40 mm
- Kv: 3700 rpm/volt

controller: Hacker Master Series 18-B-Flight
- weight: 21 g
- dimensions: 33 X 23 X 7 mm
- current drain: 18 A

battery: Kokam 2000HD (3x)
- weight: 160 g
- dimensions: 79 X 42 X 25 mm
- capacity: 2000 mAh

receiver: Schulze Alpha840W
- weight: 13.5 g
- dimensions: 52 X 21 X 13 mm
- 8 channels

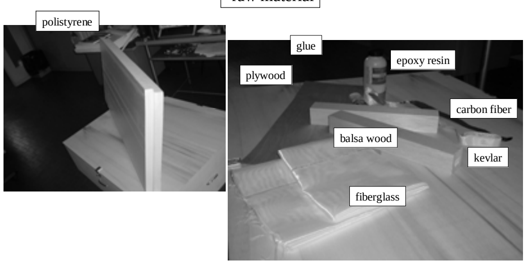
servo: Graupner C1081 (2x)
- weight: 13 g
- dimensions: 23 X 9 X 21 mm
- torque: 12 Ncm



Prototype Manufacturing - 1

IEIIT-CNR

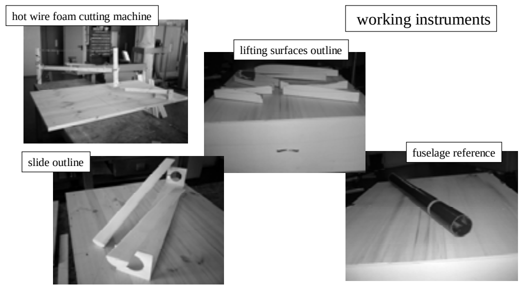
raw material



Prototype Manufacturing - 2

IEIIT-CNR

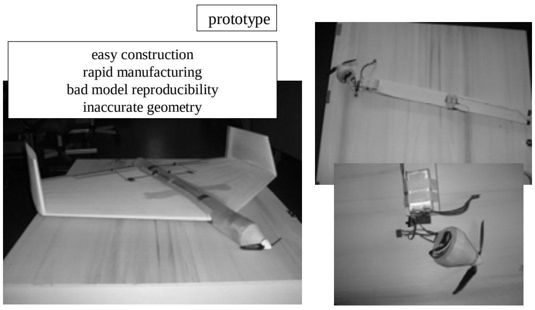
working instruments



Prototype Manufacturing - 3

IEIIT-CNR

prototype



State Space Model

IEIIT-CNR

- State space formulation obtained by linearization of the full (12 coupled nonlinear ODE) model

$$\dot{x}(t) = A(\Delta)\, x(t) + B(\Delta)\, u(t)$$

$$u(t) = -K\, x(t)$$

where $x = [V, \alpha, q, \theta]^T$ (V flight speed, α angle of attack, q and θ pitch rate and angle), Δ uncertainty

- Consider longitudinal plane dynamics stabilization
- Control u is the symmetrical elevon deflection

Uncertainty Description - 1

IEIIT-CNR

- We consider structured parameter uncertainties affecting plant and flight conditions, and aerodynamic database
- Uncertainty vector $\Delta = [\delta_1,..., \delta_{16}]$ where $\delta_i \in [\delta_i^-, \delta_i^+]$
- Key point: There is no explicit relation between state space matrices *A* and *B* and uncertainty Δ
- This is due to the fact that state space system is obtained through linearization and off-line flight simulator
- The only techniques which could be used in this case are simulation-based which lead to randomized algorithms

Uncertainty Description - 2

IEIIT-CNR

- We consider random uncertainty $\Delta = [\delta_1,..., \delta_{16}]^T$
- The pdf is either uniform (for plant and flight conditions) or Gaussian (for aerodynamic database uncertainties)
- Flight conditions uncertainties need to take into account large variations on physical parameters
- Uncertainties for aerodynamic data are related to experimental measurement or round-off errors

Plant and Flight Condition Uncertainties

IEIIT-CNR

parameter	pdf	$\bar{\delta}_i$	%	δ_i^-	δ_i^+	#
flight speed [ft/s]	U	42.65	± 15	36.25	49.05	1
altitude [ft]	U	164.04	± 100	0	328.08	2
mass [lb]	U	3.31	± 10	2.98	3.64	3
wingspan [ft]	U	3.28	± 5	3.12	3.44	4
mean aero chord [ft]	U	1.75	± 5	1.67	1.85	5
wing surface [ft^2]	U	5.61	± 10	5.06	6.18	6
moment of inertia [lb ft^2]	U	1.34	± 10	1.21	1.48	7

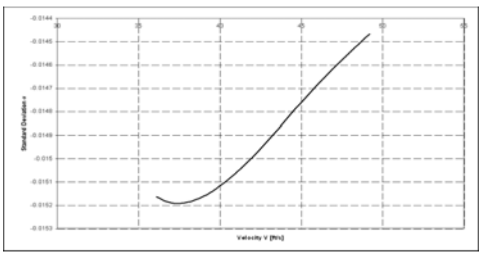
Aerodynamic Database Uncertainties

IEIIT-CNR

parameter	pdf	$\bar{\delta}_i$	σ_i	#
C_X [-]	G	-0.01215	0.00040	8
C_Z [-]	G	-0.30651	0.00500	9
C_m [-]	G	-0.02401	0.00040	10
C_{Xq} [rad^{-1}]	G	-0.20435	0.00650	11
C_{Zq} [rad^{-1}]	G	-1.49462	0.05000	12
C_{mq} [rad^{-1}]	G	-0.76882	0.01000	13
C_X [rad^{-1}]	G	-0.17072	0.00540	14
C_Z [rad^{-1}]	G	-1.41136	0.02200	15
C_m [rad^{-1}]	G	-0.94853	0.01500	16

Standard Deviation and Velocity

IEIIT-CNR



Standard deviation is experimentally computed from the velocity

Critical Parameters and Matrices

IEIIT-CNR

- We select flight speed (δ_1) and take off mass (δ_3) as critical parameters
- Flight speed is taken as critical parameter to optimize gain scheduling issues
- Take off mass is a key parameter in mission profile definition
- We define critical matrices

$$A_c^1 \quad A_c^2 \quad A_c^3 \quad A_c^4 \qquad B_c^1 \quad B_c^2 \quad B_c^3 \quad B_c^4$$

- They are constructed setting δ_1, δ_3 to the extreme values δ_1^-, δ_1^+, δ_3^-, δ_3^+ and all the remaining δ_i are equal to $\bar{\delta}_i$

IEIIT-CNR

Phase 1: Random Gain Synthesis (RGS)

- Critical parameters are flight speed and take off mass
- Specification property

  $S_1 = \{K: A_c - B_c K \textit{ satisfies the specs below}\}$

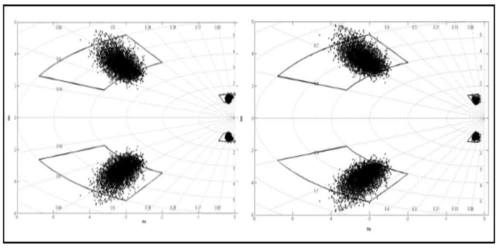
$\omega_{SP} \in [4.0,6.0]$ rad/s $\zeta_{SP} \in [0.5,0.9]$ $\omega_{PH} \in [1.0,1.5]$ rad/s

$\zeta_{PH} \in [0.1,0.3]$ $\Delta\omega_{SP} < \pm 45\%$ $\Delta\omega_{PH} < \pm 20\%$

where ω and ζ are undamped natural frequency and damping ratio of the characteristic modes; $_{SP}$ and $_{PH}$ denote short period and phugoid mode

IEIIT-CNR

Specs in the Complex Plane



IEIIT-CNR

Volume of the Good Set

- Define a bounding set B of gains K

$$B = \{K: k_i \in [k_i^-, k_i^+],\ i = 1,\ldots,4\}$$

- Define the volume of the *good set*

$$\mathrm{Vol}_{good} = \int_A \mathrm{d}\,K$$

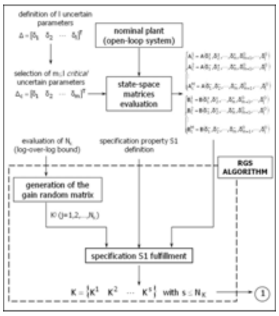
where $A = \{K \in B \cap S_1\}$

- Vol_B is simply the volume of the hyperrectangle B

IEIIT-CNR

Randomized Algorithm 1 (RGS)

- Uniform pdf for controller gains K in given intervals
- Accuracy and confidence $\varepsilon = 4 \cdot 10^{-5}$ and $\delta = 3 \cdot 10^{-4}$
- Number of random samples is computed with "Log-over-Log" Bound obtaining N = 200,000
- We obtained 5 gains K^i satisfying specification property S_1



IEIIT-CNR

Randomized Algorithm 1 (RGS)

Given ε, $\delta \in (0,1)$, RGS *returns the set of gains* $\{K^1,\ldots,K^s\}$ *satisfying* S_1

1. Compute N using the Log-over-log Bound;
2. For fixed j=1,2,…,N, generate uniformly the gain random matrix $K^j \in B$;
3. Set C=0;
4. For fixed i=1,2,3,4, compute the closed-loop matrix
   $$A_{cl}^i(K^j) = A_c^i - B_c^i K^j;$$
   - if $K^j \in S_1$, set $C = C+1$;
   - otherwise, set $C = C$;
5. End;
6. If C = 4, return the gain K^j;
7. Set $j = j+1$ and return to Step 2;
8. End

IEIIT-CNR

Random Gain Set

gain set	K_V	K_α	K_q	K_θ
K^1	0.00044023	0.09465000	0.01577400	-0.00473510
K^2	0.00021450	0.09581200	0.01555500	-0.00323510
K^3	0.00054999	0.09430800	0.01548200	-0.00486340
K^4	0.00010855	0.09183200	0.01530000	-0.00404380
K^5	0.00039238	0.09482700	0.01609300	-0.00417340

Phase 2: Random Stability Robustness Analysis (RSRA)

IEIIT-CNR

- Take $K_{rand} = K^i$ obtained in Phase 1
- Randomize Δ according to the given pdf and take N random samples Δ^i
- Specification property

$$S_2 = \{\Delta\colon A(\Delta) - B(\Delta)\ K_{rand} \textit{ satisfies the specs of } S_1\}$$

- Computation of the empirical probability of stability $\hat{p}_N$

Empirical Probability

IEIIT-CNR

- Consider fixed gain K_{rand}
- Define the probability

$$p_{true} = \int_C p(\Delta)\, d\,\Delta$$

where $C = \{\Delta \in B \cap S_2\}$ and $p(\Delta)$ is the given pdf
- Then, we introduce a "success" indicator function

$$I(\Delta^j) = 1 \text{ if } \Delta^j \in S_2$$

or $I(\Delta^j) = 0$ otherwise
- The empirical probability for S_2 is given by

$$\hat{p}_N = N_{good}/N$$

where N_{good} is equal to the number of successes

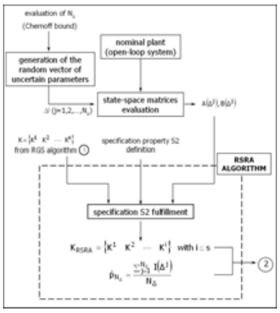
Randomized Algorithm 2 (RSRA)

IEIIT-CNR

- Take K_{rand} from Phase 1
- Accuracy and confidence

$$\varepsilon = \delta = 0.0145$$

- Number of random samples is computed with Chernoff Bound obtaining N =5,000
- Empirical probability is defined using an indicator function



Randomized Algorithm 2 (RSRA)

IEIIT-CNR

Given ε, $\delta \in (0,1)$, RSRA *returns the empirical probability* $\hat{p}_N$ *that* S_2 *is satisfied for a gain* K_{rand} *provided by Algorithm 1*

1. Compute N using the Chernoff Bound;
2. Generate N random vectors $\Delta^j \in B$ according to the given pdf;
3. For fixed j=1,2,…,N, compute the closed-loop matrix
   $$A_{cl}(\Delta^j) = A(\Delta^j) - B(\Delta^j)K_{rand};$$
   - if $A_{cl}(\Delta^j) \in S_2$, set $I(\Delta^j) = 1$;
   - otherwise, set $I(\Delta^j) = 0$;
4. End;
5. Return the empirical probability $\hat{p}_N$

Empirical Probability of Stability for Phase 2

IEIIT-CNR

gain set	empirical probability
K^1	88.56%
K^2	90.60%
K^3	89.31%
K^4	93.86%
K^5	85.14%

Probability Degradation Function

IEIIT-CNR

- Flight condition uncertainties are multiplied by the *amplification factor* $\rho > 0$ keeping the nominal value constant

$$\delta_i \in \rho\,[\delta_i^-, \delta_i^+] \quad \text{for } i = 1, 2, \ldots, 7$$

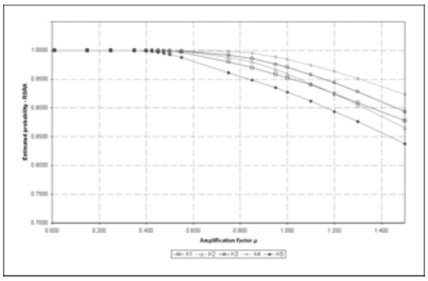
- No uncertainty affects the aerodynamic database, i.e.

$$\delta_i = \bar{\delta}_i \quad \text{for } i = 8, 9, \ldots, 16$$

- For fixed $\rho \in [0,1.5]$ we compute the empirical probability for different gain sets K^i
- The plot empirical probability vs ρ is the probability degradation function

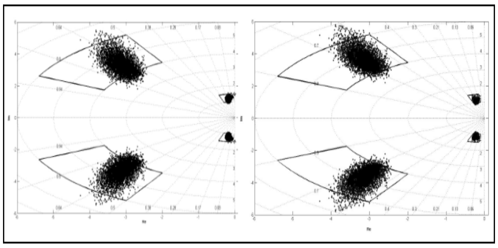
IEIIT-CNR

Probability Degradation Function for Phase 2



IEIIT-CNR

Root Locus Plot for Phase 2



Root locus for K^2 (left) and K^4 (right)

IEIIT-CNR

Phase 3: Random Performance Robustness Analysis (RPRA)

- This phase is similar to Phase 2, but military specs are considered (bandwidth criterion)
- Specification property

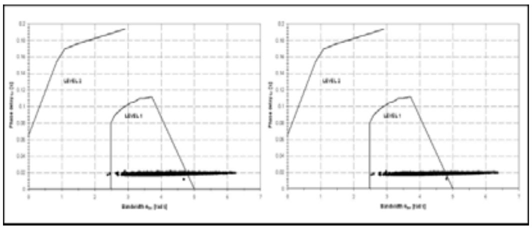
  $S_3 = \{\Delta\colon A(\Delta) - B(\Delta)\ K_{rand}$ *satisfies the specs below*$\}$

  $\omega_{BW} \in [2.5, 5.0]$ rad/s $\qquad \tau_P \in [0.0, 0.5]$ s

  where ω_{BW} and τ_P are bandwidth and phase delay of the frequency response
- Computation of the empirical probability that S_3 is satisfied

IEIIT-CNR

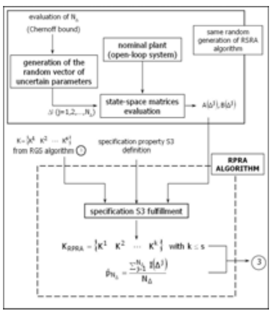
Bandwidth Criterion



IEIIT-CNR

Randomized Algorithm 3 (RPRA)

- Take K_{rand} from Phase 1
- Numer of random samples is computed with the Chernoff Bound obtaining N =5,000
- Empirical probability is defined using an indicator function



IEIIT-CNR

Randomized Algorithm 3 (RPRA)

Given N and $A_{cl}(\Delta^j)$, j=1,2,…,N, *provided by Algorithm 2,* RPRA *returns the empirical probability* $\hat{p}_N$ *that* S_3 *is satisfied for a gain* K_{rand} *provided by Algorithm 1*

1. For fixed j=1,2,…,N
   - if $A_{cl}(\Delta^j) \in S_3$, set $I(\Delta^j) = 1$;
   - otherwise, set $I(\Delta^j) = 0$;
2. End;
3. Return the empirical probability $\hat{p}_N$

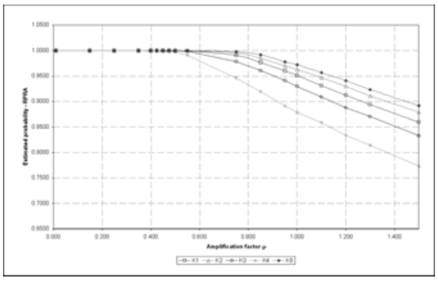
Empirical Probability of Performance for Phase 3

IEIIT-CNR

gain set	empirical probability
K^1	93.58%
K^2	95.16%
K^3	90.80%
K^4	84.78%
K^5	96.06%

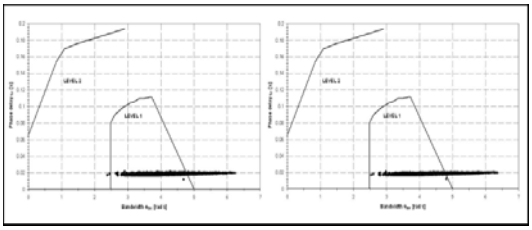
Probability Degradation Function for Phase 3

IEIIT-CNR



Bandwidth Criterion for Phase 3

IEIIT-CNR



Bandwidth criterion for K^1 (left) and K^3 (right)

Gain Selection

IEIIT-CNR

- Multi-objective criterion as a compromise between different specifications

Finally we selected gain K^1 as the best compromise between all the specs and criteria!

Conclusions: Flight Tests in Sicily - 1

IEIIT-CNR

- Evaluation of the payload carrying capabilities and autonomous flight performance
- Mission test involving altitude, velocity and heading changing was performed in Sicily
- Checking effectiveness of the control laws for longitudinal and lateral-directional dynamics
- Flight control design based on RAs for stabilization and guidance

Conclusions: Flight Tests in Sicily - 2

IEIIT-CNR

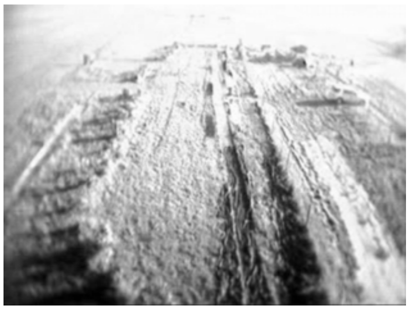
- Satisfactory response of MH1000
- Possible improvements by iterative design procedure
- Stability of the platform is crucial for the video quality and in the effectiveness of the surveillance and monitoring tasks

IEIIT-CNR

Color Camera: Right Turn

IEIIT-CNR

Color Camera: Landing Phase

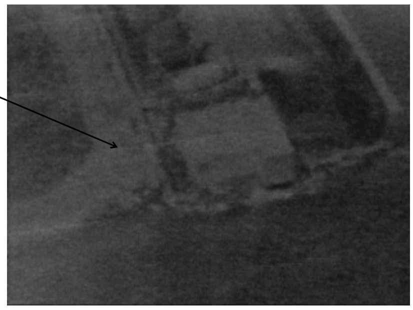


IEIIT-CNR

Infrared Camera - 1

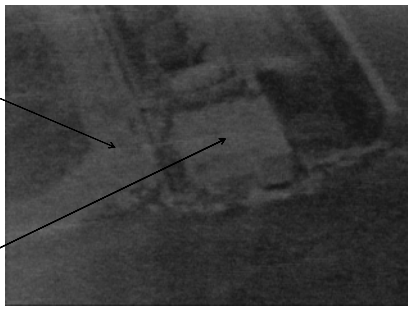
IEIIT-CNR

Infrared Camera - 1



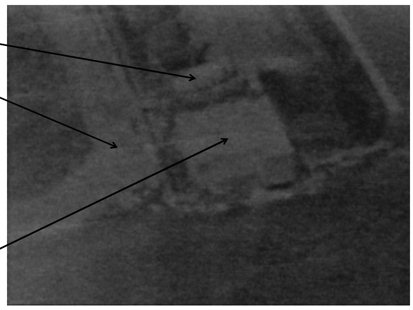
IEIIT-CNR

Infrared Camera - 1



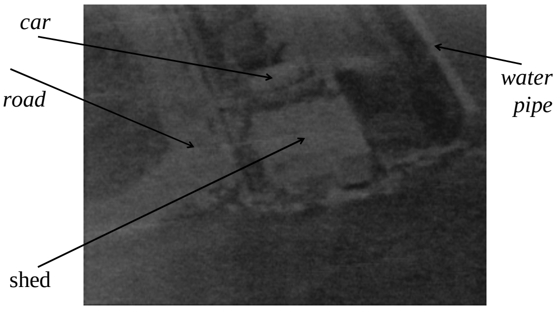
IEIIT-CNR

Infrared Camera - 1



IEIIT-CNR

Infrared Camera - 1

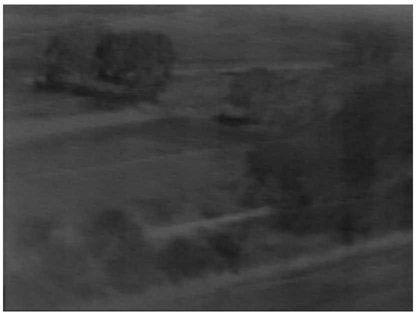


IEIIT-CNR

Infrared Camera - 2

IEIIT-CNR

Infrared Camera - 3



IEIIT-CNR

Acknowledgment

- Thanks to Andrea Sanna and his students for the flight test video and the computer graphics animation

IEIIT-CNR

Conclusions

IEIIT-CNR

PAC Algorithms

- Randomized algorithms are Probably Approximately Correct (PAC)
- We give up a guaranteed deterministic solution
- This implies accepting a “small” risk of giving a wrong solution
- The risk can be made arbitrarily small (but not zero) taking suitable values of so-called confidence and accuracy

IEIIT-CNR

PAC Algorithms

- Two open problems

- Optimization with sequential methods
- Derive “reasonable” bounds for the statistical learning theory approach